



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10312274 A

(43) Date of publication of application: 24 . 11 . 98

(51) Int. Cl.

G06F 9/06

(21) Application number: 09123832

(22) Date of filing: 14 . 05 . 97

(71) Applicant: FUJITSU LTD NIPPON TELEGR & TELEPH CORP <NTT> OKI ELECTRIC IND CO LTD

(72) Inventor: HAIZUKA TSUNEKI
TAKASAKI TORU
YAMANE KENICHI
AHIKO HIROFUMI

(54) METHOD AND DEVICE FOR SUBSTITUTING PROGRAM DURING OPERATION IN SOFTWARE SYSTEM

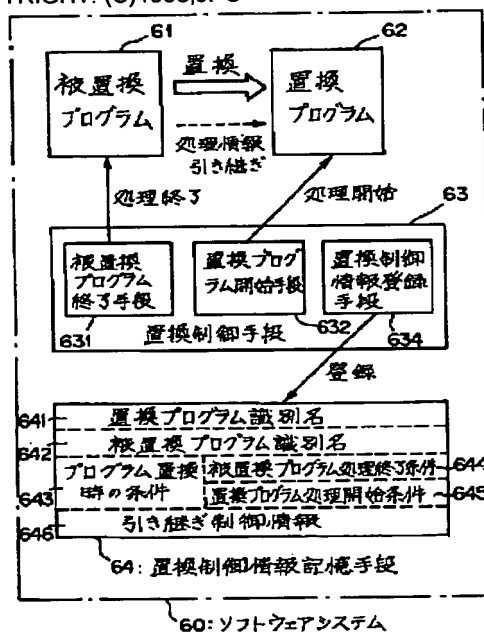
(57) Abstract:

PROBLEM TO BE SOLVED: To shorten a system processing interruption time generated with the revising work of a software system and to continue a processing before substitution even after substitution by substituting a program to be substituted for a substitution program while referring to substitution information at the time of substituting a program during an operation is substituted.

SOLUTION: A substitution control means 63 contains a substitution control information registration means 634 storing substitution control information on the substitution program 62 in a substitution control information storage means 64 when the substitution program 62 is started. At the time of substituting the program to be substituted during the operation for the substitution program 62, the substitution control information storage means 64 stores at least the program identification name 641 of the substitution program 62, the program identification name 642 of the program to be substituted 61 and a condition at the time of substituting the program. The substitution control means 63 substitutes the program to be substituted 61 for the

substitution program 62 while referring to substitution control information. Thus, the system processing interruption time is shortened.

COPYRIGHT: (C)1998,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-312274

(43) 公開日 平成10年(1998)11月24日

(51) Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 4 0

F I

G 0 6 F 9/06

5 4 0 F

審査請求 未請求 請求項の数7 OL (全 19 頁)

(21) 出願番号 特願平9-123832

(22) 出願日 平成9年(1997)5月14日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(71) 出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(74) 代理人 弁理士 小林 隆夫

最終頁に続く

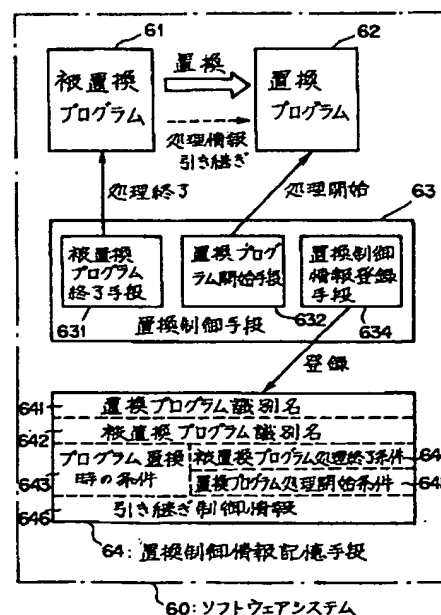
(54) 【発明の名称】 ソフトウェアシステムの稼働中プログラム置換方法および装置

(57) 【要約】

【課題】ソフトウェアシステムを稼働状態にしたままで実行中のプログラムを他のプログラムに置換する方法および装置に関し、ソフトウェアシステムの改版作業に伴って生じるシステム処理中断時間を短縮し、かつ、プログラムが置換前に行っていた処理を置換後も継続できるようにすることを目的とする。

【解決手段】少なくとも置換プログラムのプログラム識別名と被置換プログラムのプログラム識別名とプログラム置換時の条件とを置換制御情報記憶手段が記憶し、置換制御手段がこれらの置換制御情報を参照しながら、被置換プログラムの処理の終了、被置換プログラムから置換プログラムへの処理情報の引き継ぎ、置換プログラムの処理の開始を制御するように構成した。

本発明に係る原理説明図



【特許請求の範囲】

【請求項1】 それぞれ特定の機能を提供するプログラムの集合からなるソフトウェアシステムの稼働中プログラム置換方法であって、置換制御情報記憶手段と、

上記プログラムの機能とは別に設けられた置換制御手段とを備え、

該置換制御情報記憶手段は、

少なくとも、置換プログラムのプログラム識別名と、被置換プログラムのプログラム識別名と、プログラム置換時の条件とを記憶し、

該置換制御手段は、

上記の置換制御情報を参照しながら被置換プログラムを置換プログラムに置換することを特徴とするソフトウェアシステムの稼働中プログラム置換方法。

【請求項2】 該置換制御情報記憶手段は、被置換プログラムから置換プログラムへ処理情報を引き継がせるか否かを指示する引き継ぎ制御情報を更に記憶し、該置換制御手段は、プログラム置換時に、該引き継ぎ制御情報を参照することにより被置換プログラムから置換プログラムへの処理情報の引き継ぎを行うか否かを判断することを特徴とする請求項1記載のソフトウェアシステムの稼働中プログラム置換方法。

【請求項3】 該置換制御情報記憶手段は、プログラム置換時の条件として被置換プログラム処理終了条件を記憶し、

該置換制御手段は、プログラム置換時に、該被置換プログラム処理終了条件を参照することにより、被置換プログラムの処理を即時に終了させるか、あるいは、実行中の処理があればその処理が終了するのを待つか、を判断することを特徴とする請求項1または2に記載のソフトウェアシステムにの稼働中プログラム置換方法。

【請求項4】 該置換制御情報記憶手段は、プログラム置換時の条件として置換プログラム処理開始条件を記憶し、

該置換制御手段は、プログラム置換時に、該置換プログラム処理開始条件を参照することにより、被置換プログラムの処理終了時に自動的に置換プログラムの処理を開始させるか、あるいは、手動的な指示に従って置換プログラムの処理を開始させるか、を判断することを特徴とする請求項1～3のいずれかに記載のソフトウェアシステムの稼働中プログラム置換方法。

【請求項5】 それぞれ特定の機能を提供するプログラムの集合からなるソフトウェアシステムの稼働中プログラム置換装置であって、

少なくとも、置換プログラムのプログラム識別名と、被置換プログラムのプログラム識別名と、プログラム置換時の条件とを記憶する置換制御情報記憶手段と、

上記の置換制御情報を参照しながら被置換プログラムを置換プログラムに置換する置換制御手段とを備えること

を特徴とするソフトウェアシステムの稼働中プログラム置換装置。

【請求項6】 該置換制御情報記憶手段は、

上記プログラム置換時の条件として被置換プログラム処理終了条件と置換プログラム処理開始条件とを記憶し、また更に、被置換プログラムから置換プログラムへ処理情報を引き継がせるか否かを指示する引き継ぎ制御情報を記憶し、

該置換制御手段は、

該被置換プログラム処理終了条件と引き継ぎ制御情報を参照することにより、被置換プログラムの処理を終了させると共に、置換プログラムに引き継がせる処理情報があればそれを所定の記憶場所に格納する被置換プログラム終了手段と、

該置換プログラム処理開始条件と引き継ぎ制御情報を参照することにより、置換プログラムの処理を開始させると共に、被置換プログラムから引き継ぐ処理情報があればそれを所定の記憶から取り出す置換プログラム開始手段と、を含んで構成される請求項5記載のソフトウェアシステムの稼働中プログラム置換装置。

【請求項7】 該置換制御手段は、置換プログラムが起動される際に、該置換プログラムに関する置換制御情報を該置換制御情報記憶手段に記憶させる置換制御情報登録手段を含んで構成される請求項5または6に記載のソフトウェアシステムの稼働中プログラム置換装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ソフトウェアシステムを稼働状態にしたままで実行中のプログラムを他のプログラムに置換する方法および装置に関する。

【0002】例えばソフトウェアシステムの改版作業を行う際、そのソフトウェアシステムを構成するアプリケーションプログラムを旧版から新版に置換するが、それに伴って生じるプログラム処理中断時間をできるだけ短縮する必要があり、また、そのアプリケーションプログラムの処理の連続性を置換の前後で可能なかぎり維持する必要がある。

【0003】

【従来の技術】ソフトウェアシステムの改版作業を行う場合、一つの方法として、ソフトウェアシステムを停止して非稼働状態にし、ディスク上の各アプリケーションプログラムのファイルを旧版から新版に入れ替え、システム動作環境（システム立上げ時に起動するファイルの名前など）を書き換えた後、システムを再度立ち上げるというやり方がある。

【0004】しかし、この方法ではソフトウェアシステム全体の稼働を長時間中断しなければならず、ソフトウェアシステムが提供しているすべての処理をいったん中止しなければならないため問題がある。そこで、従来は、ソフトウェアシステムが稼働中に新版アプリケーション

ョンプログラムのファイルを起動し、システム動作環境に設定されている旧版のファイル名を新版のファイル名に書き換える方法などをもって処理中断時間の短縮を図っていた。

【発明が解決しようとする課題】しかし上記の方法では、旧版アプリケーションプログラムが行っていた処理を新版アプリケーションプログラムが引き継ぐことができないので、常時メッセージを受信するようなソフトウェアシステムにとっては、システムの稼働を停止して改版作業を行う場合と比べて実際上の利点はなかった。

【0005】本発明はかかる問題点に鑑みてなされたものであり、ソフトウェアシステムの改版作業に伴って生じるシステム処理中断時間を短縮し、アプリケーションプログラムが置換前に行っていた処理を置換後も継続することが可能なソフトウェアシステムにおける稼働中プログラムの置換方法および装置を提供することを目的とする。

【0006】

【課題を解決するための手段】図1は本発明に係る原理説明図である。上述の課題を解決するために、本発明においては、図1に示されるように、それぞれ特定の機能を提供するプログラムの集合からなるソフトウェアシステム60の中に、稼働中プログラムを置換する際の置換制御情報を記憶する置換制御情報記憶手段64と、置換制御情報を参照しながら被置換プログラム61を置換プログラム60に置換する置換制御手段63とを備えた稼働中プログラム置換装置を組み込んでいる。そして、置換制御情報記憶手段64は、少なくとも、置換プログラムのプログラム識別名641と、被置換プログラムのプログラム識別名642と、プログラム置換時の条件643とを記憶する。

【0007】また、本発明の稼働中プログラム置換装置において、置換制御情報記憶手段64は、上記プログラム置換時の条件643として被置換プログラム処理終了条件644と置換プログラム処理開始条件645とを記憶し、また更に、被置換プログラム61から置換プログラム62へ処理情報を引き継がせるか否かを指示する引き継ぎ制御情報646を記憶する。また、置換制御手段63は、被置換プログラム処理終了条件644と引き継ぎ制御情報646を参照することにより、被置換プログラム61の処理を終了させると共に、置換プログラム62に引き継がせる処理情報があればそれを所定の記憶場所に格納する被置換プログラム終了手段631と、置換プログラム処理開始条件645と引き継ぎ制御情報646を参照することにより、置換プログラム62の処理を開始させると共に、被置換プログラム61から引き継ぐ処理情報があればそれを所定の記憶から取り出す置換プログラム開始手段632とを含んで構成される。

【0008】また、本発明の稼働中プログラム置換装置において、置換制御手段63は、置換プログラムが起動

される際に、置換プログラムに関する置換制御情報を置換制御情報記憶手段64に記憶させる置換制御情報登録手段634を含んで構成される。

・【0009】

【作用】本発明においては、稼働中の被置換プログラム61を置換プログラム62に置換する際に、置換制御情報記憶手段64が、少なくとも、置換プログラムのプログラム識別名641と、被置換プログラムのプログラム識別名642と、プログラム置換時の条件643とを記憶し、そして置換制御手段63が、上記の置換制御情報を参照しながら被置換プログラム61を置換プログラム62に置換する。

【0010】また、本発明においては、置換制御情報記憶手段64が、被置換プログラム61から置換プログラム62へ処理情報を引き継がせるか否かを指示する引き継ぎ制御情報646を更に記憶し、そして置換制御手段63が、プログラム置換時に、引き継ぎ制御情報646を参照することにより被置換プログラム61から置換プログラム62への処理情報の引き継ぎを行うか否かを判断する。

【0011】また、本発明においては、置換制御情報記憶手段64が、プログラム置換時の条件643として被置換プログラム処理終了条件644を記憶し、そして置換制御手段63が、プログラム置換時に、被置換プログラム処理終了条件644を参照することにより、被置換プログラム61の処理を即時に終了させるか、あるいは、実行中の処理があればその処理が終了するのを待つか、を判断する。

【0012】また、本発明においては、置換制御情報記憶手段64が、プログラム置換時の条件643として置換プログラム処理開始条件645を記憶し、そして置換制御手段63が、プログラム置換時に、置換プログラム処理開始条件645を参照することにより、被置換プログラム61の処理終了時に自動的に置換プログラム62の処理を開始させるか、あるいは、手動的な指示に従って置換プログラム62の処理を開始させるかを判断する。

【実施例】以下、図面を参照して本発明の実施例を説明する。本実施例ではUNIXオペレーティングシステム上にアプリケーションシステムが構築されたソフトウェアシステムの改版を例にとって説明する。

【0013】図2は本発明が適用されるソフトウェアシステム1の構成を示している。ソフトウェアシステム1では、UNIXオペレーティングシステム2の上にn個のアプリケーションプログラムからなるアプリケーションシステム3が構築されている。ディスク4には、実行形式のアプリケーションプログラムのファイルが格納されている。アプリケーションシステム3が立ち上がる時に、それらのファイルから各アプリケーションプログラムが起動される。

【0014】上記のアプリケーションプログラムはそれぞれ特定の機能を有し、ユーザの要求に応じてその機能を実行する。ここで、ユーザとは、図中に示すように、ソフトウェアシステム1に直接接続される端末5、ネットワークを経由してソフトウェアシステム1に接続される遠隔の端末6、ネットワークを経由してソフトウェアシステム1に接続される他システム7で稼働するプログラムなどである。また、アプリケーションシステム3の内部でも、あるアプリケーションプログラムが他のアプリケーションプログラムのユーザとなりうる。以下、アプリケーションプログラムがユーザのためにその機能を実行することを「ユーザにサービスを提供する」と称する。

【0015】UNIXはマルチタスク、マルチユーザの機能を備えたオペレーティングシステムなので、アプリケーションシステム3の1つのアプリケーションプログラムが複数のユーザに対して同時に同じサービスを提供することができる。図3を参照してその仕組みについて簡単に説明する。UNIXではタスクはプロセスと呼ばれる。実行形式のアプリケーションプログラムのファイル8を起動すると、そのアプリケーションプログラムのプロセス9が生成される。そしてユーザ11がこのプロセス9に対してサービスを要求すると、それに対してプロセス9は自分の分身である子プロセス10を生成する。以下、実行形式のアプリケーションプログラムのファイル8をアプリケーションファイル、アプリケーションファイルから生成されるプロセス9をアプリケーションプロセス、アプリケーションプロセスが生成する子プロセス10をアプリケーションインスタンスと称する。本実施例で、アプリケーションプログラムとは、アプリケーションプロセスおよびそれが生成したアプリケーションインスタンスとを総称したものである。

【0016】アプリケーションプロセス9はユーザ11からのサービス要求に応じてユーザ対応にアプリケーションインスタンス10を生成する。各アプリケーションインスタンス10は、対応するユーザ11と情報の交換等を行いながら所定の処理を行い、処理終了後に消滅する。すなわち、アプリケーションプロセス9がユーザ11からのサービス要求を受け付け、アプリケーションインスタンス10がユーザ11に対してサービスを提供する。

【0017】上述のアプリケーションシステム3のアプリケーションプログラムを旧版から新版に置換しようとする場合、置換対象となるアプリケーションプログラムは次のいずれかの状態にある。

1. アプリケーションプロセスの下にアプリケーションインスタンスが生成されておらず、サービス処理は実行されていない。
2. アプリケーションプロセスの下に1つ以上のアプリケーションインスタンスが生成されており、サービス処

理が実行されている。

【0018】上記1の場合にアプリケーションプログラムを置換するには、新版のアプリケーションファイルからアプリケーションプロセスを生成し、旧版のアプリケーションプロセスを消滅させるだけなので極めて簡単である。しかし、上記2の場合は次の問題があるので置換が複雑になる。

① 旧版のアプリケーションプロセスの下に現在何個のアプリケーションインスタンスが生成されていて、それぞれがどのユーザに対応しているかを把握した上で、新版のアプリケーションプロセスの下にも同数のアプリケーションを生成し、それぞれにユーザを適正に対応付ける必要がある。

② それぞれのアプリケーションインスタンスが処理を実行中かどうか等の現在の状態を把握した上で、旧版のアプリケーションインスタンスの処理を新版のアプリケーションインスタンスに引き継がせる必要がある。

③ アプリケーションプログラムを旧版から新版へ置換している間、そのアプリケーションプログラムによるサービスについては、ユーザからのアクセスを受け付けないようにする必要がある。

【0019】本実施例のアプリケーションシステム3は、アプリケーションプログラムを旧版から新版へ置換できるように、サービス制御機能およびアプリケーションプログラム毎のサービス情報管理表を設け、サービスを一括管理できるようにしている。サービス制御機能は特許請求の範囲に規定した置換制御手段に対応し、サービス情報管理表は同じく置換制御情報記憶手段に対応する。上記の問題はサービス制御機能により以下のように解決される。

【0020】上記①の問題を解決するため、サービス制御機能は、ユーザからのサービス要求をアプリケーションプロセスに仲介し、アプリケーションインスタンスの生成と消滅の状況や、ユーザとアプリケーションインスタンスとの対応関係を把握する。

【0021】また、上記②の問題を解決するため、サービス制御機能は、生成されたアプリケーションインスタンス毎にサービス状態管理表を割り当てる。サービス状態管理表にはそれに対応するサービスの現在の状態を示す情報が格納され、サービス制御機能はサービス状態管理表を参照することによりそのサービスを実行しているアプリケーションインスタンスの状態を把握することができる。

【0022】また、上記③の問題を解決するために、ユーザとアプリケーションプロセスまたはアプリケーションインスタンスとの間のアクセスはすべてサービス制御機能を介して行われるようにし、アプリケーションプログラム置換中、サービス制御機能はそのアプリケーションプログラムについてのユーザからのアクセスを受け付けないようにする。

【0023】次に、サービス制御機能によるアプリケーションプログラムの置換の概要を図4～図6を参照して説明する。置換手順は次の5つに大別できる。

- ① 新版アプリケーションプログラムの起動
- ② サービスの閉塞
- ②' 旧版アプリケーションプログラムから新版アプリケーションプログラムへの処理情報の引き継ぎ
- ③ サービスの閉塞解除
- ④ 旧版アプリケーションプログラムの停止

【0024】以下、上記の各手順の概要を順に説明する。

① 新版アプリケーションプログラムの起動
置換前の状態では、図4(a)に示すように、旧版アプリケーションプログラム22がユーザにサービスを提供し、旧版アプリケーションプロセス12の下にいくつかのアプリケーションインスタンス14が存在し、旧版アプリケーションプロセス12にはサービス情報管理表13が、各アプリケーションインスタンス14にはサービス状態管理表15がそれぞれ割り当てられている。そこで、ディスクに格納されている新版アプリケーションファイルから新版アプリケーションプログラム23を起動することにより新版アプリケーションプロセス16を生成し、それに対するサービス情報管理表17をアプリケーションシステムに登録し、更に、旧版と同数のアプリケーションインスタンス18を生成してそれらに対してサービス状態管理表19を割り当てる。

【0025】② サービスの閉塞
新版アプリケーションプログラム23が起動された時点では、図4(b)に示すように、旧版アプリケーションプログラム22と新版アプリケーションプログラム23が同時に存在している。ただし、旧版アプリケーションプログラム22はサービス制御機能を介してユーザ20との間で情報交換等のアクセスを行いながらサービスを提供する「活動状態」にあるのに対して、新版アプリケーションプログラム23はサービスの提供を行っていない「非活動状態」にある。

【0026】旧版アプリケーションプログラム22を新版アプリケーションプログラム23に置換するには、旧版アプリケーションインスタンス18が現在実行している処理を中断し、ユーザ20から旧版アプリケーションプログラム22へのアクセスを受け付けなくにする必要がある。これはユーザ20の側から見れば、その時まで利用していたサービスが突然利用できなくなることを意味する。このようにユーザへのサービスの提供を中断することを、以後、「サービスを閉塞する」と称する。図5(c)に示すように、サービスを閉塞することによりアプリケーションプログラムは旧版、新版とも「非活動状態」になる。

【0027】②' 旧版アプリケーションプログラムから新版アプリケーションプログラムへの処理情報の引き継

ぎ

次に、図5(d)に示すように、旧版アプリケーションインスタンス14がそれまで実行していた処理の情報21を、それぞれに対応する新版アプリケーションインスタンス18に引き継がせる。

【0028】③ サービスの閉塞解除

次に、このサービスを利用しているユーザ20が新版アプリケーションプログラム23との間でアクセスを行えるようにし、旧版アプリケーションインスタンス14が中断した処理を新版アプリケーションインスタンス18に再開させる。このようにユーザへのサービスの提供を再開することを、以後、「サービスを閉塞解除する」と称する。図6(e)に示すように、サービスを閉塞解除することにより新版アプリケーションプログラム23は「活動状態」になり、旧版アプリケーションプログラム22は「非活動状態」になる。

【0029】④ 旧版アプリケーションプログラムの停止

最後に、旧版アプリケーションインスタンス14を消滅させ、旧版アプリケーションに対するサービス状態管理表15の割当てを解除し、更に旧版アプリケーションプロセス12を消滅させることにより旧版アプリケーションプログラム22を停止する。

【0030】ところで、アプリケーションインスタンスの状態に着目すれば、上記の②～③の過程で以下のことを配慮することが望まれる。上記の「活動状態」のアプリケーションプログラムにおいて、サービスを実行しているアプリケーションインスタンスには図7に示す2つの状態がありうる。ここで、「処理実行状態」とはアプリケーションインスタンスが本来の処理を継続的に実行している状態を意味し、「アイドル状態」とはアプリケーションインスタンスが事象の発生や外部からの指示などを待っているか、あるいは何らかの理由で本来の処理を休止している状態を意味する。

【0031】サービスの閉塞を開始した時に旧版アプリケーションインスタンス14が「処理実行状態」であり、かつその処理は時間的な中断が不可能もしくは困難な性質のものである場合、その処理を中断してすぐに置換を始めてしまうと置換の前後で処理の連続性が失われてしまうことがある。このような場合、旧版アプリケーションインスタンス14の実行中の処理が終了するのを待ってから置換を開始するのが望ましい。この場合は、当然、新版アプリケーションインスタンス18が旧版アプリケーションインスタンス14の処理情報21を引き継ぐ必要はない。一方、サービスの閉塞を開始した時に旧版アプリケーションインスタンス14が「アイドル状態」である場合は、即時に旧版アプリケーションインスタンス14の処理を中断して置換を開始しても問題はないが、新版アプリケーションインスタンス18は中断した処理を再開するために、旧版アプリケーションインス

タンス14がそれまで使用していた処理情報21を引き継ぐ必要がある。

【0032】図8は本実施例のサービス制御機能、サービス情報管理表、およびサービス状態管理表を一層具体的に示している。サービス制御機能30は、図8(a)の機能構成が示すように、サービス情報登録機能31とサービス情報登録解除機能32とサービス閉塞機能33とサービス閉塞解除機能34とサービス起動機能35とサービス停止機能36とサービス情報管理表40とサービス状態管理表50とで構成されている。

【0033】これらの機能は、システム保守者が対応するコマンドをコンソールから投入することによって起動される場合と、実行中のプログラムがその機能呼び出すことにより起動される場合と、サービス制御機能30が自らの判断で起動する場合とがある。

【0034】サービス情報登録機能31は、アプリケーションシステムに新しいアプリケーションプログラムを追加する際にそれに対応するサービス情報管理表40に登録したり、既存のアプリケーションプログラムに対するサービス情報管理表40の内容を変更したりする機能である。サービス情報登録機能31は請求項7で規定した置換制御情報登録手段に対応する。

【0035】サービス情報登録解除機能32は、アプリケーションシステムから既存のアプリケーションプログラムを削除する際にそれに対応するサービス情報管理表40の登録を解除する機能である。

【0036】サービス閉塞機能33は、旧版アプリケーションプログラムから新版アプリケーションプログラムへの置換を行う際に、旧版アプリケーションインスタンスが現在実行している処理を中断し、ユーザから旧版アプリケーションインスタンスへのアクセスを受け付けないようにすることにより、ユーザへのサービスの提供を中断する機能である。サービス閉塞機能33は請求項6で規定した被置換プログラム処理終了手段に対応する。

【0037】サービス閉塞解除機能34は、サービス閉塞前に旧版アプリケーションプログラムによるサービスを利用していたユーザが新版アプリケーションインスタンスとの間でアクセスを行えるようにし、旧版アプリケーションインスタンスが中断した処理を新版アプリケーションインスタンスに再開させることにより、ユーザへのサービスの提供を再開する機能である。サービス閉塞機能34は請求項6で規定した置換プログラム処理開始手段に対応する。

【0038】サービス起動機能35は、ディスクに格納されているアプリケーションファイルを起動してアプリケーションプロセスを生成することにより、それに対応するサービスの提供を開始するために主にアプリケーションシステム立上げ時に使用する機能である。アプリケーションプログラムの置換時には、新版アプリケーションプログラムを起動するために使用できる。

【0039】サービス停止機能36は、アプリケーションインスタンスとアプリケーションプロセスを消滅させることにより、それに対応するサービスの提供を停止するために主にアプリケーションシステム立下げ時に使用する機能である。アプリケーション置換時には、旧版アプリケーションプログラムを停止するために使用できる。

【0040】図8(b)はサービス情報管理表40の内容を示している。サービス情報管理表40は特許請求の範囲で規定した置換制御情報記憶手段に対応する。サービス情報管理表40には、アプリケーションファイル名41、旧版アプリケーションファイル名42、閉塞条件フラグ43、および閉塞解除条件44が格納される。これらの情報は特許請求の範囲で規定した置換制御情報と以下のように対応している。

アプリケーションファイル名41 → 置換プログラム識別名

旧版アプリケーションファイル名42 → 被置換プログラム識別名

閉塞条件フラグ43 → 被置換プログラム処理終了条件と引き継ぎ制御情報

閉塞解除条件フラグ44 → 置換プログラム処理開始条件

【0041】本実施例では、置換制御情報記憶手段の一実施形態として、サービス情報管理表をアプリケーションプログラムごとに登録し、それぞれのサービス情報管理表はそれに対応するアプリケーションプログラムに関する置換制御情報を保持する。これについて図9を参照して簡単に説明する。図9はアプリケーションプログラムAをアプリケーションプログラムBに置換し、その後、更にアプリケーションプログラムBをアプリケーションプログラムCに置換する場合の例を示している。

【0042】アプリケーションプログラムAからBへの置換では、アプリケーションプログラムAに対するサービス情報管理表のアプリケーションファイル名、閉塞条件フラグと、アプリケーションプログラムBに対するサービス情報管理表のアプリケーションファイル名、旧版アプリケーションファイル名、閉塞解除条件フラグが置換制御情報として参照される。そして、アプリケーションプログラムBからCへの置換では、アプリケーションプログラムBに対するサービス情報管理表のアプリケーションファイル名、閉塞条件フラグと、アプリケーションプログラムCに対するサービス情報管理表のアプリケーションファイル名、旧版アプリケーションファイル名、閉塞解除条件フラグが置換制御情報として参照される。

【0043】すなわち、各アプリケーションプログラムに対するサービス情報管理表には、そのアプリケーションプログラムが置換プログラムであるときの情報と被置換プログラムであるときの情報が格納されている。そし

て、アプリケーションプログラムの置換時に、置換プログラムと被置換プログラム双方のサービス情報管理表が置換制御情報記憶手段となり、その場合に必要で置換制御情報だけがそれぞれのサービス情報管理表から読み出される。

【0044】次に、図8(b)のサービス情報管理表40の各情報について説明する。アプリケーションファイル名41は、サービス情報管理表40が割り当てられているアプリケーションプログラムに対するアプリケーションファイル名である。

【0045】旧版アプリケーションファイル名42は、サービス情報管理表40が割り当てられているアプリケーションプログラムが置換プログラムである場合の被置換プログラム、すなわち旧版アプリケーションプログラムに対するアプリケーションファイル名である。サービス情報管理表40が割り当てられているアプリケーションプログラムが被置換プログラムである場合はこの情報は参照されない。

【0046】閉塞条件フラグ43は、アプリケーションプログラムを置換する際のサービス閉塞条件を「即時閉塞」または「待ち処理終了後閉塞」で指示するフラグである。サービス情報管理表40が割り当てられているアプリケーションプログラムが被置換プログラム（すなわち旧版アプリケーションプログラム）である場合に閉塞条件フラグ43が参照される。

【0047】閉塞条件フラグ43が「即時閉塞」を指示する場合、サービス閉塞機能33を起動すると即時にサービスが閉塞される。閉塞条件フラグ43が「待ち処理終了後閉塞」を指示する場合、サービス閉塞機能33を起動すると、旧版アプリケーションインスタンスが「処理実行状態」と「アイドル状態」（図7を参照）のどちらであるかが検査され、「処理実行状態」であればその処理の終了後にサービスが閉塞され、「アイドル状態」であれば即時にサービスが閉塞される。

【0048】なお、閉塞条件フラグ43は、特許請求の範囲で規定した引き継ぎ制御情報も兼ねている。これは、サービスを即時に閉塞するか、あるいは現在実行中の処理が終了してからサービスを閉塞するかの条件が、処理情報を旧版アプリケーションプログラムから新版アプリケーションプログラムに引き継ぐか否かの条件に対応しているためである。すなわち、旧版アプリケーションインスタンスが「アイドル状態」（図7を参照）であれば、サービスを即時に閉塞でき、それまでの処理情報の引き継ぎを行うが、旧版アプリケーションインスタンスが「処理実行状態」であれば、その処理が終了するのを待ってからサービスを閉塞し、処理は終了しているので処理情報の引き継ぎは行わない。したがって、閉塞条件フラグ43が「即時閉塞」を指示する場合は処理情報の引き継ぎを行い、「待ち処理終了後閉塞」を指示する場合は処理情報の引き継ぎを行わない。

【0049】システム保守者は、サービスの特性を考慮して、次の基準で閉塞条件フラグ43の指示を決定することができる。

① 開始してから短時間で処理を終えて終了するようなサービスは一般に「アイドル状態」の時間が短いので、このようなサービスに対しては「待ち処理終了後閉塞」を指示する。

② 開始してから長時間（例えば1年以上）かかるようなサービスは一般に「アイドル状態」の時間が長いので、このようなサービスに対しては「即時閉塞」を指示する。

【0050】閉塞解除条件フラグ44は、アプリケーションプログラムを置換する際のサービス閉塞解除条件を「自動解除」と「手動解除」で指示するフラグである。サービス情報管理表40が割り当てられているアプリケーションプログラムが置換プログラム（すなわち新版アプリケーションプログラム）である場合に閉塞解除条件フラグ44が参照される。

【0051】閉塞解除条件フラグ44が「自動解除」を指示する場合、サービス閉塞機能33によりサービスが閉塞され、旧版アプリケーションプログラムの処理が終了した時点で、自動的にサービス閉塞解除機能34が起動されてサービスが閉塞解除される。閉塞解除条件フラグ44が「手動解除」を指示する場合、サービス閉塞機能33によりサービスが閉塞された後、システム保守者等が手動で指示する（例えば、サービス閉塞解除のコマンドをキーボードから入力する）ことによってサービス閉塞解除機能34が起動されてサービスが閉塞解除される。

【0052】図8(c)はサービス状態管理表50の内容を示している。サービス状態管理表50は、各アプリケーションインスタンスが実行しているサービスの状態に関する情報を保持する管理表であり、生成されたアプリケーションインスタンスに対して割り当てられ、そのアプリケーションインスタンスが消滅すると割り当てが解除される動的な情報である。サービス状態管理表50の割り当てと割り当て解除はサービス制御機能30により自動的に行われる。

【0053】サービス状態管理表50には、アプリケーションファイル名51、アプリケーションインスタンス識別子52、サービス状態フラグ53、アプリケーションインスタンス状態フラグ54、引き継ぎ情報アドレス55、および引き継ぎ情報語数56が格納される。

【0054】次に、図8(c)のサービス状態管理表50の各情報について説明する。アプリケーションファイル名51は、そのサービスを実行するアプリケーションプログラムに対するアプリケーションファイル名である。アプリケーションインスタンス識別子52は、サービス状態管理表50に対応するアプリケーションインスタンスに付与された識別子である。

【0055】サービス制御機能は、サービス情報管理表40のアプリケーションファイル名41、サービス状態管理表50のアプリケーションファイル名51、アプリケーションインスタンス識別子52を参照することにより、アプリケーションプロセスとアプリケーションインスタンスとの対応関係を認識し、アプリケーションインスタンスを識別することができる。

【0056】サービス状態フラグ53は、アプリケーションファイル名51に対応するアプリケーションプログラムが実行するサービスの状態を「提供中」または「閉塞中」で指示するフラグである。「提供中」とはユーザに対してサービスを提供している状態であり、「閉塞中」とはサービス閉塞機能33の実行によりサービスが閉塞されている状態である。

【0057】アプリケーションインスタンス状態フラグ54は、サービス状態管理表50が割り当てられたアプリケーションインスタンスの現在の状態が「処理実行状態」と「アイドル状態」（図7を参照）のどちらであるかを示すフラグである。

【0058】引き継ぎ情報アドレス55と引き継ぎ情報語数56は、旧版アプリケーションプログラムから新版アプリケーションプログラムへの処理情報の引き継ぎを行う際の作業用情報である。本実施例では、アプリケーションプログラムの置換時に、旧版アプリケーションプログラムがその処理情報をいったんディスク上の所定の記憶場所に保存し、新版アプリケーションプログラムがディスクに保存された処理情報を読み出すという形で処理情報の引き継ぎを行う。したがって、引き継ぎ情報アドレス55は処理情報が保存されているディスク上の先頭アドレスであり、引き継ぎ情報語数56はディスクに保存された処理情報の語数である。

【0059】次に、図4～図6で概要を示したアプリケーションプログラムの置換手順を、図8で具体化したサービス制御機能、サービス情報管理表、サービス状態管理表の内容に即して詳細に説明する。

【0060】図10は、アプリケーションプログラムの置換の過程における旧版アプリケーションプログラム22と新版アプリケーションプログラム23の稼働状態の例およびサービス情報管理表とサービス状態管理表の内容を示している。すなわち、旧版アプリケーションプロセス12の下にn個の旧版アプリケーションインスタンス14が生成され、一方、新版アプリケーションプロセス16の下にn個の新版アプリケーションインスタンス18が生成されている。また、旧版アプリケーションプログラム22に対してサービス情報管理表13が登録され、n個の旧版アプリケーションインスタンス14のそれぞれにサービス状態管理表15が割り当てられ、一方、新版アプリケーションプログラム23に対してサービス情報管理表17が登録され、n個の新版アプリケーションインスタンス18のそれぞれにサービス状態管理

表19が割り当てられている。また、便宜上、旧版アプリケーションファイル名はOLD、旧版アプリケーションインスタンス識別子はO1、O2、・・・Onとし、新版アプリケーションファイル名はNEW、新版アプリケーションインスタンス識別子はN1、N2、・・・Nnとする。以下の説明では図10に示した符号を使用する。

【0061】以下、前述したアプリケーションプログラム置換手順①～④の詳細を図10～図14を参照して説明する。

【0062】① 新版アプリケーションプログラムの起動

システム保守者は、予め作成された新版アプリケーションファイルNEWをディスクに格納しておき（ステップ1）、次に、新版アプリケーションファイルNEWから新版アプリケーションプログラム23を起動するためにサービス起動のコマンドをコンソールから投入する（ステップ2）。

【0063】サービス起動機能が起動されて新版アプリケーションプロセス17が生成される（ステップ3）。この時、新版アプリケーションプログラム23に対してサービス情報管理表が既に登録されているかどうかを確認するために現在登録されているサービス情報管理表が検索される（ステップ4）。もし新版アプリケーションプログラム23に対するサービス情報管理表が未登録であれば、サービス情報登録機能が起動され、新版アプリケーションプログラム23に対してサービス情報管理表17が登録される（ステップ5）。ステップ4と5の処理を新版アプリケーションプログラム23の中に作りこんでおけば、新版アプリケーションプログラム23の起動時に自動的にそれに対するサービス情報管理表17を登録することができる。このように新版アプリケーションプログラムが起動時に自らサービス情報登録機能を呼び出す方法は請求項7に対応する。

【0064】次に、新版アプリケーションプログラム23に対するサービス情報管理表17の旧版アプリケーションファイル名171が参照され、その結果、アプリケーションファイル名OLDに対応する旧版アプリケーションプログラム22から、アプリケーションファイル名NEWに対応する新版アプリケーションプログラム23への置換が行われることがサービス制御機能により認識される（ステップ6）。そして、旧版アプリケーションプログラム22に対するサービス情報管理表13に関連するサービス状態管理表15が検索され、その結果、旧版アプリケーションプロセス12の下にn個のアプリケーションインスタンスが存在することが確認される。サービス情報管理表13とサービス状態管理表15との関連はアプリケーションファイル名131とアプリケーションファイル名151との一致を調べることで確認できる（ステップ7）。

【0065】上記のステップ6と7の処理は、サービス制御機能が行っても、新版アプリケーションプロセス16が自分で行ってもよい。ステップ7の処理の結果を受けて、新版アプリケーションプロセス16の下にn個のアプリケーションインスタンスN1～Nnが生成される（ステップ8）。更に、生成された新版アプリケーションインスタンスN1～Nnのそれぞれに対してサービス状態管理表19が割り当てられる（ステップ9）。

【0066】㊸ サービスの閉塞（旧版アプリケーションプログラムからの処理情報の引き継ぎを含む）
新版アプリケーションプログラム23の起動が終了すると、次にサービス閉塞機能が起動される（ステップ10）。サービス閉塞機能の起動は、新版アプリケーションプログラム23の起動終了後にサービス制御機能が自動的に行う方法と、システム保守者がサービス閉塞のコマンドをコンソールから投入することによって行う方法とが可能である。

【0067】サービス閉塞機能が起動されると、旧版アプリケーションプログラム22に対するサービス情報管理表13の閉塞条件フラグ133が検査される（ステップ11）。閉塞条件フラグ133が「待ち処理終了後閉塞」を指示していれば、待ち処理（ステップ12）を開始する。閉塞条件フラグ133が「即時閉塞」を指示していれば、待ち処理は行われない。この場合の待ち処理とは、n個の旧版アプリケーションインスタンス14がすべて「アイドル状態」になるのを待つ処理のことである。サービス制御機能は旧版アプリケーションインスタンス14に対するサービス状態管理表154のアプリケーションインスタンス状態フラグ154を監視し、n個の旧版アプリケーションインスタンス14がすべて「アイドル状態」になれば待ち処理を終了する。

【0068】各アプリケーションインスタンスは、自分の状態をサービス制御機能に通知するために、サービス状態管理表のアプリケーションインスタンス状態フラグをセットまたはリセットする。したがって、アプリケーションプログラムの中にアプリケーションインスタンス状態フラグを適宜にセット・リセットする処理を作りこんでおく必要がある。

【0069】待ち処理（ステップ12）の終了後に、あるいは閉塞条件フラグ133が「即時閉塞」を指示している場合は即時に、サービス制御機能は、n個の旧版アプリケーションインスタンスのサービス状態管理表15のサービス状態フラグ153を「提供中」から「閉塞中」に変え（ステップ13）、n個の旧版アプリケーションインスタンス14の実行中の処理を終了させ（ステップ14）、旧版アプリケーションプログラム22に対するユーザからのアクセスを禁止する（ステップ15）。例えば、旧版アプリケーションプログラム22に対してユーザからのアクセスがあってもそのユーザに拒否応答を返す。

【0070】サービスが閉塞されると、処理情報の引き継ぎを行うか否かを判断するために、旧版アプリケーションプログラム22に対するサービス情報管理表13の閉塞条件フラグ133が検査される（ステップ16）。閉塞条件フラグ133が「待ち処理終了後閉塞」を指示している場合、前述したように処理情報の引き継ぎの必要はないので次のステップ17と18は省略される。一方、閉塞条件フラグ133が「即時閉塞」を指示している場合、旧版アプリケーションインスタンス14から新版アプリケーションインスタンス18に処理情報を引き継がせるために次のステップ17と18が実行される。

【0071】各旧版アプリケーションインスタンス14がサービス閉塞直前まで実行していた処理の情報がディスクに保存される（ステップ17）。そして、保存された処理情報のディスク上の格納アドレスと語数が、各旧版アプリケーションインスタンス14に対応する新版アプリケーションインスタンス18のサービス状態管理表19の引き継ぎ情報アドレス195と引き継ぎ情報語数196に書き込まれる（ステップ18）。ステップ16～18の処理情報引き継ぎの処理は、サービス制御機能が行っても、各旧版アプリケーションインスタンス14が自分で行ってもよい。

【0072】ステップ18までの処理が終了した旧版アプリケーションインスタンス14は、サービス閉塞のための処理が完了したことを示すために、各自のサービス状態管理表15のサービス状態フラグ153の指示を「閉塞中」にする（ステップ19）。

【0073】㊹ サービスの閉塞解除（新版アプリケーションプログラムへの処理情報の引き継ぎを含む）
サービス制御機能は、サービスの閉塞の完了を判断するために、旧版アプリケーションインスタンス14のサービス状態管理表15のサービス状態フラグ153を監視する（ステップ20）。n個の旧版アプリケーションインスタンス14のサービス状態フラグ153がすべて「閉塞中」を指示すれば、サービスの閉塞が完了したと判断される。

【0074】サービス制御機能は、サービスの閉塞を自動解除するか、あるいは手動解除するかを判断するために、新版アプリケーションプログラム23に対するサービス情報管理表17の閉塞解除条件フラグ174を検査する（ステップ21）。閉塞解除条件フラグ174が「手動解除」を指示していれば、サービス制御機能は待ち処理（ステップ22）を行い、閉塞解除条件フラグ174が「自動解除」を指示していれば待ち処理を行わない。この場合、待ち処理とは、システム保守者がサービスの閉塞解除のコマンドを投入するのを待つことである。サービスの閉塞解除のコマンドが投入されれば以下のステップに進む。

【0075】旧版アプリケーションインスタンス14の処理情報がディスクに保存されているか否かを判断する

ために、旧版アプリケーションプログラム22に対するサービス情報管理表13の閉塞条件フラグ133が検査される(ステップ23)。閉塞条件フラグ133が「即時閉塞」を指示している場合、保存されている旧版アプリケーションインスタンス14の処理情報を新版アプリケーションインスタンス18が引き継ぐための処理(ステップ24)が行われ、閉塞条件フラグ133が「待ち処理終了後閉塞」を指示している場合、前述したように処理情報は保存されていないので引き継ぎ処理は行われない。

【0076】新版アプリケーションインスタンス18のサービス状態管理表19の引き継ぎ情報アドレス195が示すディスク上のアドレスから、引き継ぎ情報語数196が示す語数の処理情報が読み出される(ステップ24)。ステップ23~24の処理は、サービス制御機能が行っても、各新版アプリケーションインスタンス18が自分で行ってもよい。

【0077】ステップ24までの処理が終了した新版アプリケーションインスタンス18は、サービス閉塞解除の準備が完了したことを示すために、各自のサービス状態管理表19のサービス状態フラグ193の指示を「提供中」にする(ステップ25)。

【0078】サービス制御機能は、サービス閉塞解除の準備完了を判断するために、新版アプリケーションインスタンス18のサービス状態管理表19のサービス状態フラグ193を監視する(ステップ26)。n個の新版アプリケーションインスタンス18のサービス状態フラグ193がすべて「提供中」を指示すれば、サービスの閉塞解除の準備が整ったと判断される。

【0079】そして、サービス制御機能は、新版アプリケーションプログラム23に対するユーザからのアクセスを許可する(ステップ27)。例えば、サービス閉塞前にそのサービスを利用していたユーザに対してサービス再開が通知され、それらのユーザは、サービス閉塞前にアクセスしていた旧版アプリケーションインスタンス14に対応する新版アプリケーションインスタンス18にアクセスできるようになる。以後、新版アプリケーションプログラム23は通常の動作を開始し、ユーザにサービスを提供する。

【0080】④ 旧版アプリケーションプログラムの停止

システム保守者は、新版アプリケーションプログラム23によるサービスが正常に再開したことを確認した後、サービス停止のコマンドを投入する(ステップ28)。サービス停止機能が起動されることにより、n個の旧版アプリケーションインスタンス14に対するサービス状態管理表19の割当てがそれぞれ解除され、次いで旧版アプリケーションインスタンス14がすべて消滅し、最後に旧版アプリケーションプロセス12が消滅する(ステップ29)。

【0081】更に、システム保守者がサービス情報登録解除のコマンドを投入する(ステップ30)と、旧版アプリケーションプログラム22に対するサービス情報管理表13の登録が解除される(ステップ31)。以上で、旧版アプリケーションプログラム22から新版アプリケーションプログラム23への置換が完了する。

【0082】

【発明の効果】以上説明したように、本発明によれば、ソフトウェアシステムの改版時のアプリケーションプログラムの処理を一括管理することにより、稼働中のアプリケーションプログラムを旧版から新版へ効率的に置換し、それに伴って生じる処理中断時間を短縮することができる。

【0083】また、本発明によれば、置換前に旧版アプリケーションプログラムが行っていた処理の情報を置換後に新版アプリケーションプログラムに引き継がせることで、置換のために旧版アプリケーションプログラムが中断した処理を新版アプリケーションプログラムに再開させることができる。

【0084】また、本発明によれば、置換するアプリケーションプログラムの処理の特性に応じてアプリケーションプログラム置換時の条件を指定できるので、置換の前後のアプリケーションプログラムの処理の連続性が高いレベルで保証される。

【0085】また、本発明によれば、新版アプリケーションプログラムをソフトウェアシステムにおいて起動する際にそれに対する置換制御情報が自動的に登録されるので、改版作業を省力化することができる。

【図面の簡単な説明】

【図1】本発明に係る原理説明図である。

【図2】実施例のソフトウェアシステムの構成を示す図である。

【図3】実施例のアプリケーションプログラムの稼働の概念図である。

【図4】実施例のアプリケーションプログラムの置換手順の概要を説明するための図である。

【図5】実施例のアプリケーションプログラムの置換手順の概要を説明するための図である。

【図6】実施例のアプリケーションプログラムの置換手順の概要を説明するための図である。

【図7】実施例のアプリケーションインスタンスの状態の遷移を説明するための図である。

【図8】実施例のサービス制御機能の構成を示す図である。

【図9】置換時のサービス情報管理表の参照情報を示す図である。

【図10】置換時の旧版および新版アプリケーションプログラムの稼働状態の例を示す図である。

【図11】実施例のアプリケーションプログラムの置換手順の詳細を説明するフローチャートである。

【図12】実施例のアプリケーションプログラムの置換手順の詳細を説明するフローチャートである。

【図13】実施例のアプリケーションプログラムの置換手順の詳細を説明するフローチャートである。

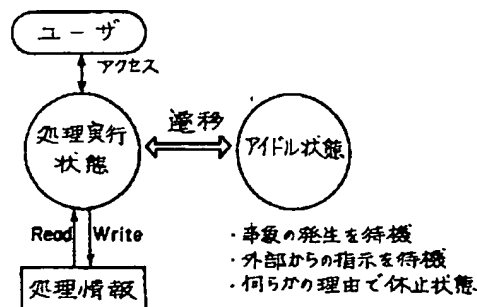
【図14】実施例のアプリケーションプログラムの置換手順の詳細を説明するフローチャートである。

【符号の説明】

- | | |
|-----------------------------------|----------------------------------|
| 1 ソフトウェアシステム | ス情報管理表 |
| 2 UNIXオペレーティングシステム | 18 新版アプリケーションインスタンス |
| 3 アプリケーションシステム | 19 新版アプリケーションインスタンスに対するサービス状態管理表 |
| 4 ディスク | 21 アプリケーションインスタンスの処理情報 |
| 5 ソフトウェアシステム1に直接接続される端末 | 22 旧版アプリケーションプログラム |
| 6 ネットワーク経由でソフトウェアシステム1に接続される遠隔の端末 | 24 新版アプリケーションプログラム |
| 7 ネットワーク経由でソフトウェアシステム1に接続される他システム | 30 サービス制御機能 |
| 8 アプリケーションファイル | 31 サービス情報登録機能 |
| 9 アプリケーションプロセス | 32 サービス情報登録解除機能 |
| 10 アプリケーションインスタンス | 33 サービス閉塞機能 |
| 11、20 ユーザ | 34 サービス閉塞解除機能 |
| 12 旧版アプリケーションプロセス | 35 サービス起動機能 |
| 13 旧版アプリケーションプログラムに対するサービス情報管理表 | 36 サービス停止機能 |
| 14 旧版アプリケーションインスタンス | 40、13、17 サービス情報管理表 |
| 15 旧版アプリケーションインスタンスに対するサービス状態管理表 | 41、131、171 アプリケーションファイル名 |
| 16 新版アプリケーションプロセス | 42、172 旧版アプリケーションファイル名 |
| 17 新版アプリケーションプログラムに対するサービス情報管理表 | 43、133 閉塞条件フラグ |
| | 44、174 閉塞解除条件フラグ |
| | 50、15、19 サービス状態管理表 |
| | 51、151、191 アプリケーションファイル名 |
| | 52、152、192 アプリケーションインスタンス識別子 |
| | 53、153、193 サービス状態フラグ |
| | 54、154、194 アプリケーションインスタンス状態フラグ |
| | 55、195 引き継ぎ情報アドレス |
| | 56、196 引き継ぎ情報語数 |

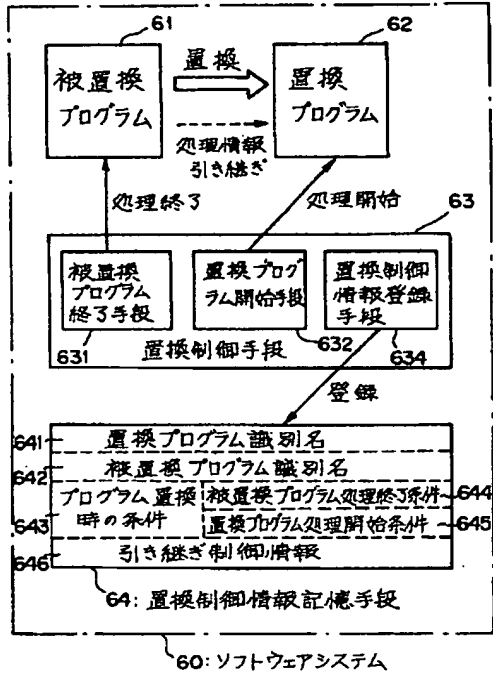
【図7】

アプリケーションインスタンスの状態遷移



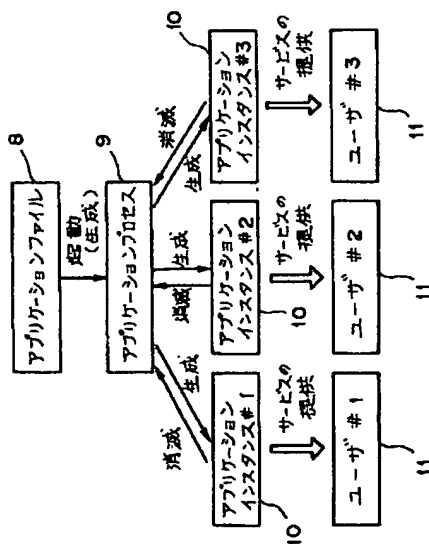
【図1】

本発明に係る原理説明図



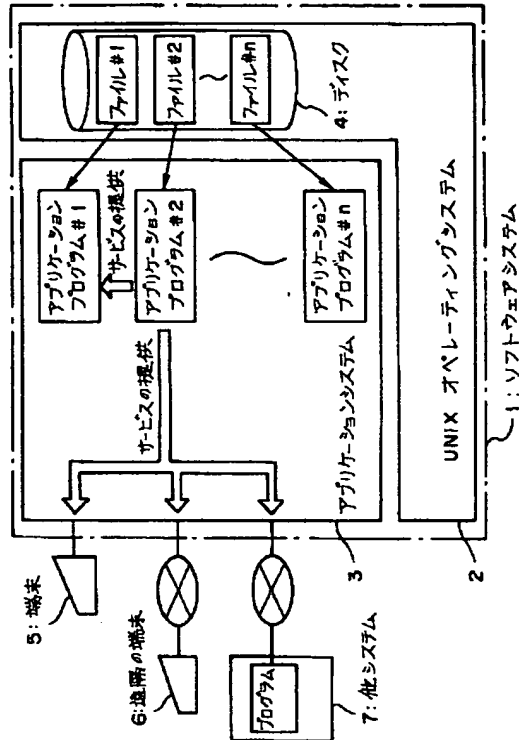
【図3】

アプリケーションプログラムの稼働の概念図

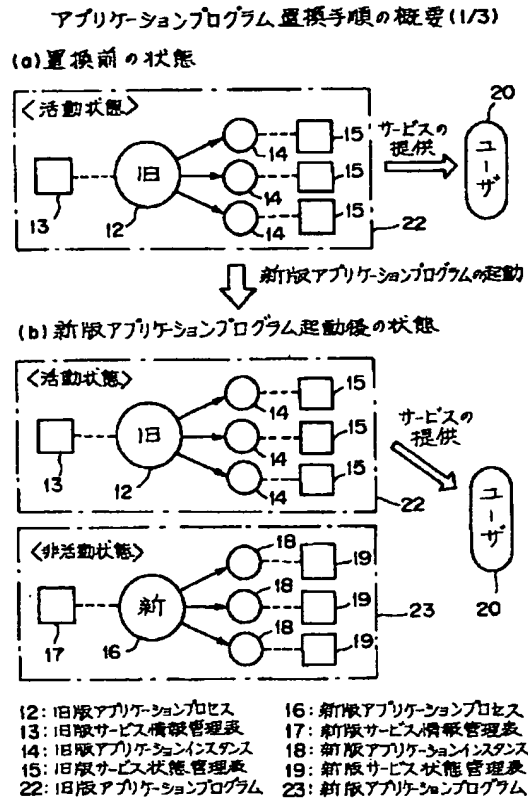


【図2】

実施例のソフトウェアシステムの構成

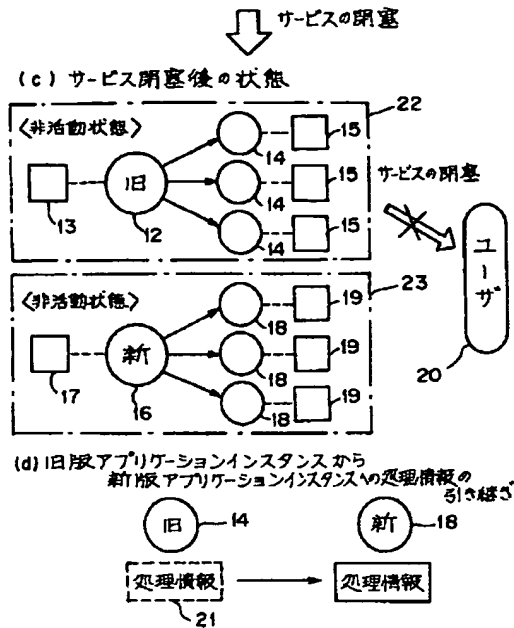


【図4】



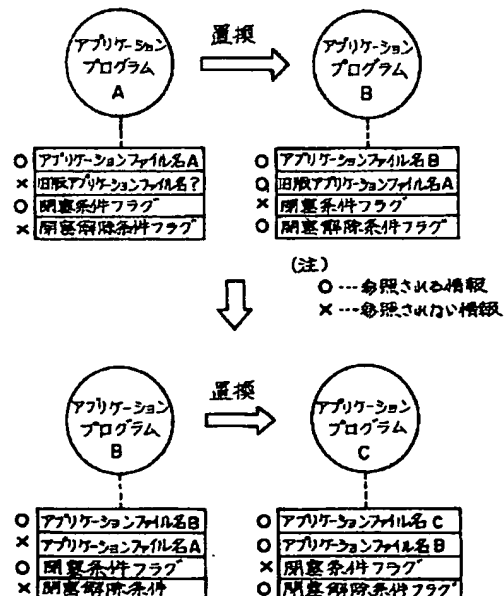
【図5】

アプリケーションプログラム置換手順の概要(2/3)



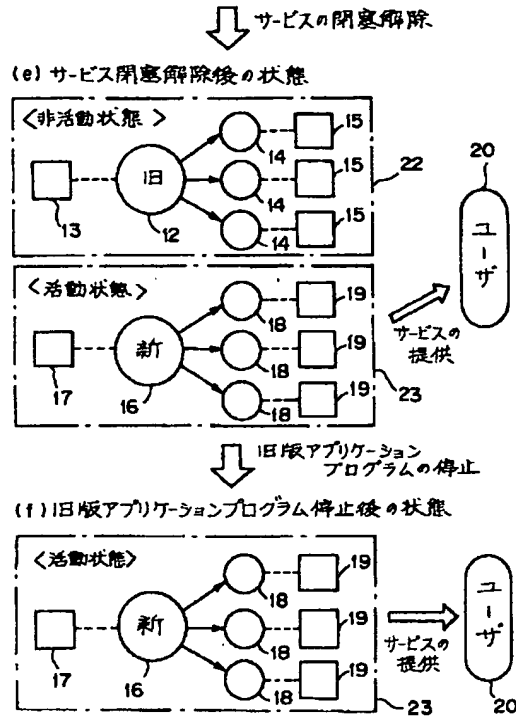
【図9】

置換時のサービス情報管理表の参照情報



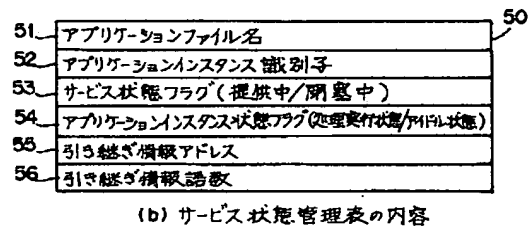
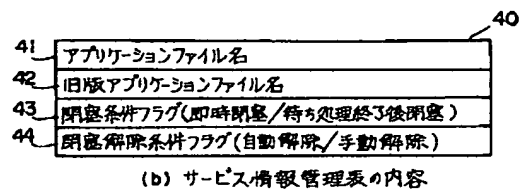
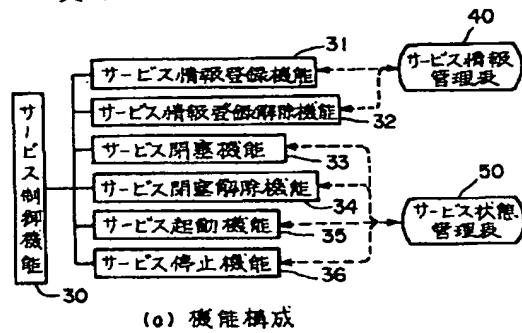
【図6】

アプリケーションプログラム置換手順の概要(3/3)



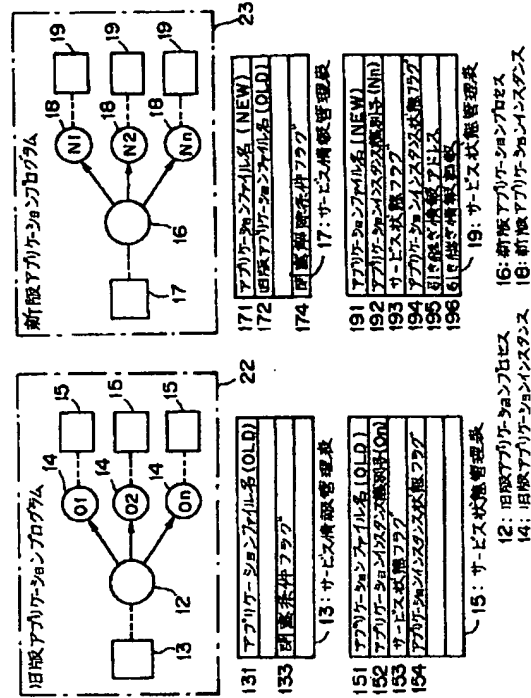
【図8】

実施例のサービス制御機能の構成



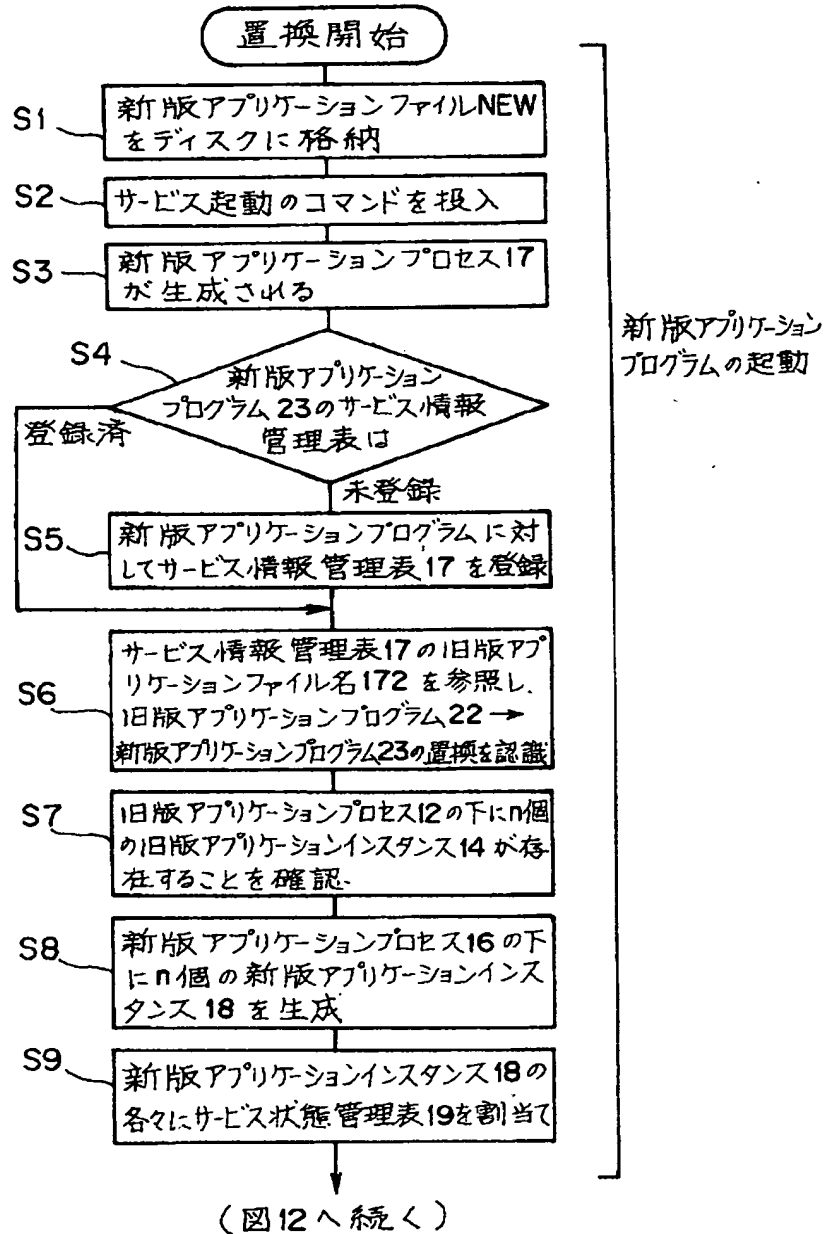
【図10】

置換時のアプリケーションプログラムの稼動状態の例



【図11】

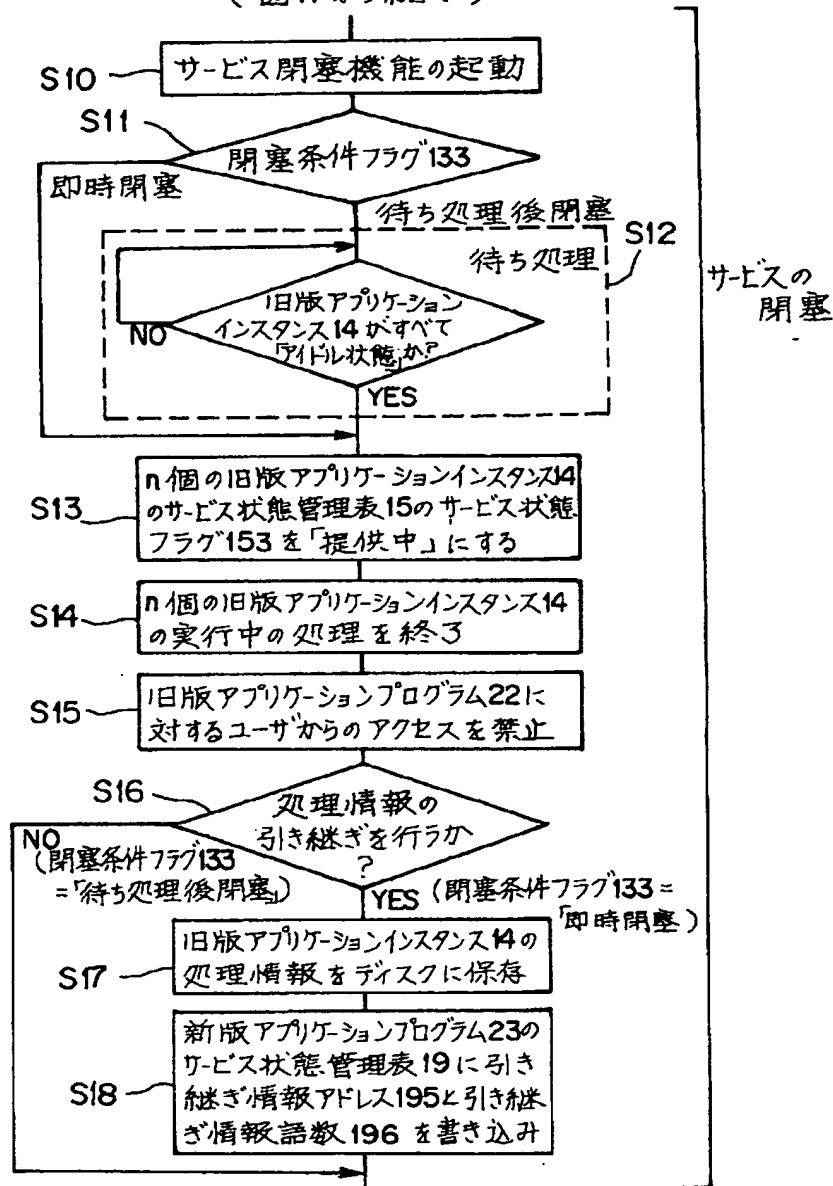
アプリケーションプログラム置換手順(1/4)



【図12】

アプリケーションプログラム置換手順(2/4)

(図11 から続く)

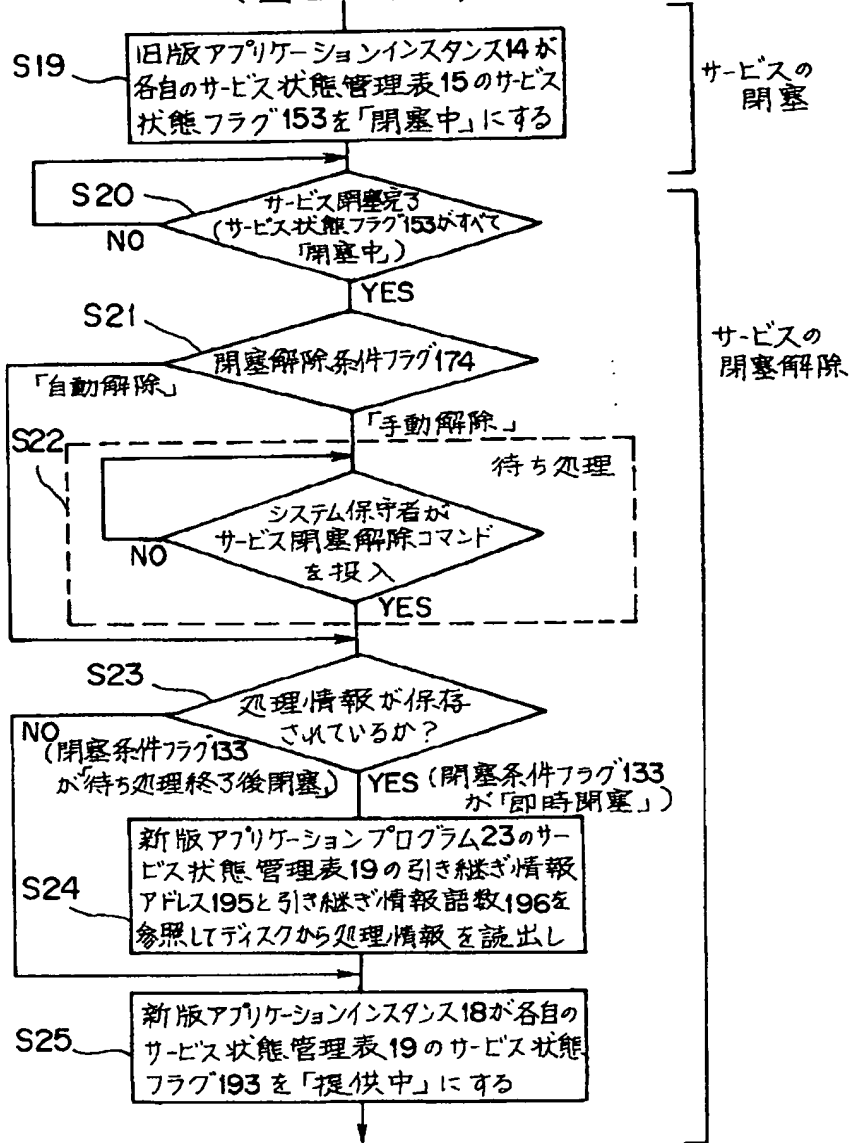


(図13へ続く)

【図13】

アプリケーションプログラム置換手順(3/4)

(図12から続く)

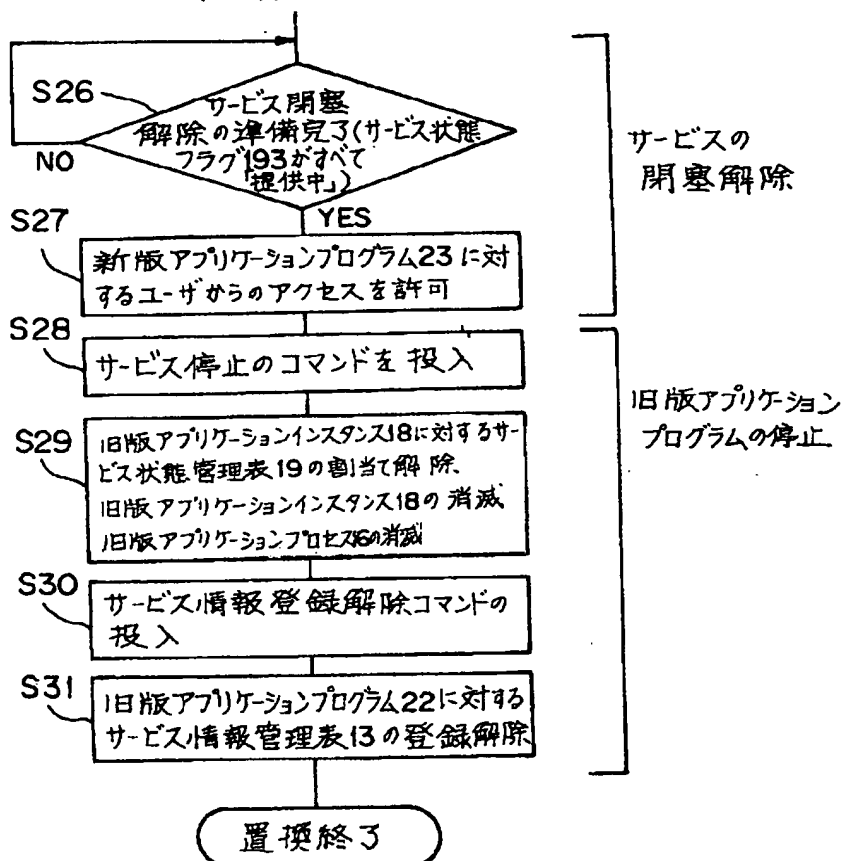


(図14へ続く)

【図14】

アプリケーションプログラム置換手順(4/4)

(図13 から続く)



フロントページの続き

(72)発明者 灰▲塚▼ 凡樹
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 高崎 徹
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 山根 健一
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 阿彦 浩文
東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内